

# **OBJECT-ORIENTED PROGRAMMING USING C++**

## **ASSIGNMENT – UNIT I**

### **Paper Code: CIC-211**

1. Explain the Object-Oriented Programming paradigm and discuss its major benefits.
2. What are tokens in C++? Explain keywords, identifiers, and constants with suitable examples.
3. Explain the different data types in C++ and discuss type compatibility.
4. Explain variables and the different categories of operators available in C++.
5. What is implicit type conversion? Explain with suitable examples.
6. Explain operator precedence in C++ and discuss its importance in evaluating expressions.
7. Explain the main() function and function prototyping in C++.
8. Differentiate between call by value and call by reference. Explain call by reference with an example.
9. What are inline functions and friend functions? Explain their purpose and characteristics.
10. Explain function overloading and default parameter values in C++ with suitable examples.
11. Write a C++ program to demonstrate call by reference by swapping two numbers using a function.
12. Write a C++ program using an inline function to calculate the square of a number.
13. Write a C++ program to demonstrate function overloading for calculating the area of different geometric shapes.
14. Write a C++ program using a friend function to access and display private data members of a class.
15. Write a C++ program to demonstrate the use of default parameter values in a function.

# **OBJECT-ORIENTED PROGRAMMING USING C++**

## **ASSIGNMENT – UNIT II**

### **Paper Code: CIC-211**

1. Explain how a class is specified in C++. Discuss the role of member functions.
2. Explain encapsulation and information hiding. How are they related to abstract data types?
3. Differentiate between a class and an object. Explain the identity and behaviour of an object.
4. What are static member functions? Explain their characteristics and use.
5. Explain arrays of objects in C++ with a suitable example.
6. What are constructors and destructors? Explain their purpose and differences.
7. Explain parameterized constructors and copy constructors with suitable examples.
8. What is a dynamic constructor? Explain its purpose in C++.
9. Explain dynamic memory allocation and explicit type conversions in C++.
10. Explain operator overloading in C++. Discuss its purpose and significance.
11. Write a C++ program to define a class Student with data members for roll number and marks, and display the details using member functions.
12. Write a C++ program to demonstrate the use of a parameterized constructor for initializing class data members.
13. Write a C++ program to demonstrate a copy constructor by creating a copy of an existing object.
14. Write a C++ program to dynamically allocate memory for an array of integers and display its elements.
15. Write a C++ program to overload an operator for a user-defined class.

# **OBJECT-ORIENTED PROGRAMMING USING C++**

## **ASSIGNMENT – UNIT III**

### **Paper Code: CIC-211**

1. Explain inheritance in C++ and discuss the different inheritance methods.
2. What is a class hierarchy? Explain the concept of derivation using public, private, and protected inheritance.
3. Differentiate between aggregation and inheritance with suitable examples.
4. Explain inheritance constructors and the role of initialization lists.
5. Differentiate between composition and classification hierarchies.
6. Explain containership in C++ with a suitable example.
7. What is polymorphism? Explain the major categories of polymorphic techniques.
8. Explain polymorphism by parameter and parametric polymorphism.
9. What is a generic function? Explain the concept of a template function in C++.
10. Explain function overriding, run-time polymorphism, and virtual functions.
11. Write a C++ program to demonstrate single inheritance using a base class and a derived class.
12. Write a C++ program to demonstrate multilevel inheritance and display information from all levels.
13. Write a C++ program to demonstrate function overriding using a base class and a derived class.
14. Write a C++ program using a virtual function to demonstrate run-time polymorphism.
15. Write a C++ program to implement a generic function template that finds the larger of two values.

# **OBJECT-ORIENTED PROGRAMMING USING C++**

## **ASSIGNMENT – UNIT IV**

### **Paper Code: CIC-211**

1. Explain the concept of standard C++ classes and their significance.
2. What is multiple inheritance? Explain how it is implemented in C++.
3. Explain persistent objects and their role in C++ programming.
4. Explain streams and files in C++. Discuss their basic purpose.
5. What are namespaces in C++? Explain their need and use.
6. Explain exception handling in C++ and discuss its importance.
7. What are generic classes? Explain their role in generic programming.
8. Explain the organization of the Standard Template Library (STL) and discuss standard containers.
9. Explain algorithms, function objects, iterators, and allocators in the context of STL.
10. Explain strings, streams, manipulators, user-defined manipulators, and vectors in C++.
11. Write a C++ program to demonstrate multiple inheritance using two base classes and one derived class.
12. Write a C++ program to write data to a file and then read the data from the file using C++ streams.
13. Write a C++ program to demonstrate exception handling for division by zero.
14. Write a C++ program using a vector to store a set of integers and display its elements using an iterator.
15. Write a C++ program using an STL algorithm to sort a collection of integers and display the sorted result.